

METHOD FOR ENCRYPTION KEY GENERATION

Field of the Invention

The invention relates to transmission of data over an unsecured interface, and in particular to a method for generating an encryption key for encrypting
5 plaintext then later recreates the encryption key for decryption of the data.

Problem

It is a problem in the field of encrypting data for transmission and storage across an unsecured interface to prevent unauthorized devices from intercepting and decrypting the transmitted data while also providing an encryption key that can
10 be recreated by the encrypting device to later decrypt the stored data without storing the encryption key within the encrypting device.

Reading and writing digital content across an unsecured interface to a storage device exposes the content to possible duplication and theft of information. Data that can be read and understood without any special measures is called
15 plaintext. The method of disguising plaintext in such a way as to hide its message is called encryption. Encrypting plaintext results in unreadable gibberish called ciphertext. Encryption is used to ensure that information is hidden from anyone for whom it is not intended, including those who can see the encrypted data. The process of reverting ciphertext back to its original plaintext is called decryption.
20 Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables the storage of sensitive information or the transmission of the information across an insecure network so that it cannot be read by anyone except the intended recipient.

A cryptographic algorithm, or cipher, is a mathematical function used in the
25 encryption and decryption process. A cryptographic algorithm works in combination

with a key – a word, number, or phrase – to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. Therefore, the security of the encrypted data is dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

5 There are two types of encryption. Conventional encryption, also called secret-key or symmetric-key encryption, where one key is used for both encryption and decryption. Another encryption system, public key cryptography, is an asymmetric scheme that uses a pair of keys for encryption: a public key to encrypt the message and a corresponding private key to decrypt the encrypted message.

10 Conventional encryption is fast and is useful for encrypting data that isn't going anywhere. However, a problem with the use conventional encryption for encrypting data that is being transmitted over an insecure interface can be quite expensive due to the difficulty of secure key distribution.

For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are at different physical locations, they must distribute the key via some secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the secret key in transit can later read, modify, and forge all information encrypted or authenticated with that secret key. The persistent problem with conventional encryption is key distribution: how to get the key to the recipient without someone intercepting it.

Pretty Good Privacy (PGP)

A know public encryption system is the PGP, which is a hybrid cryptosystem. PGP first compresses the plaintext for two reasons. First compression saves modern transmission time and disk storage space and, more importantly, it

strengthens the cryptographic security. Attackers exploit patterns found in plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing the resistance to attack. Compression within PGP is a one-way hash function which takes a variable length plaintext message and produces a fixed-length hashed value. Hash functions have been used in the computer science industry for a long time. A hash function is a function, mathematical or otherwise, that takes a variable length digital input string and converts it to a fixed length digital output string called a hashed value.

PGP then creates a session key which is a one-time-only secret key randomly generated. The session key along with a conventional encryption algorithm is used to encrypt the plaintext. Once the plaintext is encrypted, the session key is encrypted to the recipient's private key. The public key-encrypted session key is transmitted along with the ciphertext to the recipient. The recipient uses his private key to recover the temporary session key, which is then used to decrypt the conventionally-encrypted ciphertext.

The combination of the two encryption methods combines the convenience of public key encryption with the speed of conventional encryption. Transmitting the public key-encrypted session over an insecure interface renders the PGP encryption system subject to a man-in-the-middle attack. It is possible for an attacker to post a phony public key with the name and identification of the recipient. Data encrypted to the recipient is received by the attacker, the message is now in the wrong hands. Using conventional encryption systems, it is vital that the sender insure that the public key being used to encrypt the session key does in fact belong to the recipient.

Digital Signature Standard (DSS)

Another public encryption system is the digital signature standard (DSS). The security of DSS is dependent on maintaining the secrecy of users' private keys. Users must therefore guard against the unauthorized acquisition of their private
5 keys. The DSS standard specifies general security requirements for generating digital signatures. Digital signatures are used to detect unauthorized modification to data and to authenticate the identity of the signatory. In addition, the recipient of the signed data can use a digital signature in proving to a third party that the signature was in fact generated by the signatory.

10 Like PGP, DSS uses a secure hash algorithm in conjunction with a digital signature algorithm (DSA) to generate a secure signature for a document and to verify the signature of the received document. The DSA is used by the signatory to generate a digital signature and by the verifier to verify the authenticity of the signature. Each signatory has a public and a private key. The private key is used
15 in the signature generation process and the public key is used in the signature verification process. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key. For both signature generation and verification, the data which is referred to as a message is reduced by means of the secure hash algorithm. An adversary who does not know the
20 private key of the signatory, cannot generate the correct signature of the signatory. In other words, signatures cannot be forged. However, by using the signatory's public key, anyone can verify a correctly signed message.

While the DSS standard just described provides a method for generating a signature from a private signatory key, the method fails to provide a means for
25 protecting the private signatory key. Instead, DSS is dependent on maintaining the

secrecy of the users' private key. Users must therefore guard against the unauthorized acquisition of their private keys. Another problem associated with the public key system is that the public and the private keys are mathematically related. Given enough time and computing power, the private key can be derived from the public key.

For these reasons, a need exists for a method creating an encryption key that can be reproduced at a later date for decrypting the data without saving the encryption key on the encrypting device or with the transmitted ciphertext.

Solution

The present method for encryption key generation overcomes the problems outlined above and advances the art by providing a method of combining the speed of conventional encryption with the security of public key encryption. The host device encrypting the plaintext to be transmitted over the unsecured interface is assigned a host identification. The host identification is stored in a secure location within the host device.

The host identification is analogous to the private key. Only the host device can generate the encryption key used to later decrypt the ciphertext. A second variable, a content identification, is generated by the host device. Each successive block of plaintext to be encrypted uses a different content identification. The host identification along with the content identification is used for generating an encryption key to encrypt a block of plaintext. This second variable, the content identification, is analogous to the public key. The content identification is transmitted with the resulting ciphertext and together the ciphertext and content identification are stored for retrieval at a later time.

The encryption key is generated following a method that can be repeated later using the same host identification and content identification to generate the same encryption key. In other words, the formula used to generate the encryption key is deterministic. In an embodiment all combinations of the host identification and the content identification are concatenated to generate the encryption key. Following the same method in reverse using the retrieved content identification in conjunction with host identification generates the same combinations. Concatenating the same combinations in the same order produces the same encryption key for decrypting the ciphertext.

In an alternative embodiment, a time variable is also used to generate the encryption key. In this embodiment, the time variable provides a method for generating an encryption key to encrypt plaintext that must be retrieved and deciphered within a specific time period. When the specific time period has elapsed, the time variable used to generate the encryption key will have changed. Thus, generating a different encryption key. In this embodiment, decryption of the ciphertext is for a limited time only.

Brief Description of the Drawings

Figure 1 illustrates a block schematic diagram of a host device for use with the method for encryption key generation;

Figure 2 illustrates combinations of the host identification and content identification used to generate the encryption key;

Figure 3 illustrates combination of the host identification, content identification, and time used to generate the encryption key in an alternative embodiment;

Figure 4 illustrates a flow diagram for encrypting plaintext using the present method for encryption key generation; and

Figure 5 illustrates a flow diagram for decrypting ciphertext using the present method for encryption key generation.

5

Detailed Description

The invention summarized above and defined by the enumerated claims may be better understood by referring to the following detailed description, which should be read in conjunction with the accompanying drawings. This detailed description of the preferred embodiment is not intended to limit the enumerated claims, but to serve as a particular example thereof. In addition, the phraseology and terminology employed herein is for the purpose of description, and not of limitation.

Reading and writing digital content across an unsecured interface to a storage device exposes the content to possible duplication and theft of information. Data that can be read and understood without any special measures is called plaintext. The method of disguising plaintext in such a way as to hide its message is called encryption. Encrypting plaintext results in unreadable gibberish called ciphertext. Encryption is used to ensure that information is hidden from anyone for whom it is not intended, including those who can see the encrypted data. The process of reverting ciphertext back to its original plaintext is called decryption. Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables the storage of sensitive information or the transmission of the information across an insecure network so that it cannot be read by anyone except the intended recipient.

A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key – a word, number, or phrase – to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. Therefore, the security of the encrypted data is dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

There are two types of encryption. Conventional encryption where one key is used for both encryption and decryption and public key cryptography, an asymmetric scheme that uses a pair of keys for encryption: a public key to encrypt the message and a corresponding private key to decrypt the encrypted message. The present method for encryption key generation provides a method for generating an encryption key for use with a conventional encryption system wherein the key can later be recreated for use in decrypting the ciphertext. Typically, conventional encryption is fast and therefore useful for encrypting data that isn't going anywhere. However, a problem with the use conventional encryption for encrypting data is the difficulty of secure key distribution.

Using the present method for encryption key generation, an encryption key is generated wherein only a portion of the encryption key is distributed with the ciphertext. The other portion of the encryption key remains with the host device that generated the encryption key. Thus, only the host device that encrypted the data has the information necessary to recreate the encryption key to decrypt the resulting ciphertext. The method combines conventional and public key cryptography. One portion of the encryption key is analogous to the public key and transmitted with the ciphertext while the portion of the key that remains with the

encryption device is analogous to the private key. Like conventional cryptography, the same key that is used to encrypt the data is used to decrypt the data.

Thus, the present method for encryption key generation allows businesses that transmit secure data over an unsecured interface for storage at another location to encrypt the data for transmission, transmit the ciphertext with a portion of the encryption key, then later retrieve the ciphertext and recreate the encryption key to decrypt the ciphertext. The combination of the two encryption methods combines the convenience of public key encryption with the speed of conventional encryption. Used together, the present method for encryption key generation improves performance and encryption key distribution.

Encryption Key Generation—Figure 1:

The present method for encryption key generation uses a pseudo public key and pseudo private key. In this embodiment, the public key is a content identification number and the private key is a host identification.

Referring to figure 1, the host device 100 generating the encryption key includes host identification 110 stored in a secure location within the host device, thus resembling a private key. The private portion of the key, the host identification, is unique to the device, therefore generating an encryption key that cannot be generated by a host device having a different host identification. The public portion of the encryption key is the content identification.

The content identification is a unique identification that is generated by host device 100. Each block of data to be transmitted is assigned a unique content identification. The unique content identification can be a randomly generated code, can be created sequentially or another method of setting the content identification could be substituted. Other known methods for generating a content identification

include randomly selecting an initial content identification code and incrementing the content identification for transmission of successive blocks or the initial content identification could be derived from a protocol such as Realtime Transport Protocol (RTP). Those skilled in the art will appreciate that alternative methods of generating a content identification can be substituted.

Encryption and Transmission—Figures 2 and 3:

For each block of plaintext that is to be transmitted across an unsecured interface, a content identification is generated. Using the host identification and the content identification, the host device generates an encryption key having the following properties. First, the host device generates an encryption key containing each possible combination of host identification and content identification. Referring to figure 2, a first combination 210 is host identification 202 followed by content identification 204. A second combination 220 is content identification 204 followed by host identification 202. The formula for generating the encryption key may concatenate the first combination followed by the second combination to produce a longer encryption key 230, 240. Encryption key size is measured in bits. In this example, a one-byte host identification combined with a one-byte content identification results in an encryption key of four bytes. Increasing the size of the host identification and/or the content identification results in a larger key size. In public key encryption, the larger the key, the more secure the ciphertext.

The encryption key could also be generated from an eight-byte host identification and an eight-byte content identification. In this example, the first combination 210 is exclusive ORed with second combination 220 using modulo 256 arithmetic calculations. Thus, producing an eight-byte encryption key that is more secure. Those skilled in the art will recognize that alternative methods of

coalescing the host identification and the content identification may be substituted to generate the encryption key. Concatenating or exclusive ORing the host identification and the content identification are for illustration and not intended as a limitation.

5 Whichever method is followed to generate the encryption key from a combination of the host identification and the content identification, the same method is used to generate all encryption keys. Using the same method to combine the host identification and the content identification to generate the encryption key results in an encryption key that is deterministic. In other words, 10 using the same host identification and the same content identification to generate the encryption key will always produce the same encryption key.

Generating an encryption key using a host identification provides a method for preventing another device from decrypting the ciphertext. If another device recovered the content identification appended to the ciphertext, the encryption key 15 generated by that device would combine the host identification and the content identification to generate the encryption key. Since the host identification is different, the encryption key generated would be different even if the same method of generating the encryption key were followed.

In an alternative embodiment, a third variable is included with the host 20 identification and the content identification to generate the encryption key. In this embodiment, time is the third variable and the time is produced by secure clock 120 within the host device 100 shown in figure 1. Referring to figure 3, adding the third variable of time produces six unique combinations 310, 320, 330, 340, 350 and 360. Using the example where each variable, host identification 202, content 25 identification 204 and time 206, are each one-byte in length, concatenation of the

six combinations produces an eighteen-byte encryption key. As discussed previously, increasing the size of the host identification, content identification and/or the time variable can increase the length of the encryption key.

Adding the third variable of time increases the security of the encryption key.

5 For each subsequent block of plaintext to be transmitted over the unsecured interface, the content identification can be incremented and a new time variable used. In this example the time variable is the time when the encryption key is generated. Using a new time variable to generate a new encryption key provides a method for increasing the security of the encryption key and thus the resulting
10 ciphertext. Changing the content identification and the time variable for each successive block of plaintext provides a method for generating a unique encryption key for each successive block of plaintext.

Encryption and Storage of Plaintext—Figures 1 and 4:

Referring to the flow diagram in figure 4, as previously discussed, in block
15 410 a unique content identification is generated by the host device. For each successive block of plaintext to be transmitted, either a new content identification is created in block 410 or the previous content identification is incremented or otherwise modified in block 420. Using host identification 202 and the content identification from block 420, an encryption key is generated. In an embodiment
20 previously discussed, the encryption key is a concatenation of all combinations of the host identification and the content identification. In an alternative embodiment, time variable 206 is also used to generate the unique encryption key in block 430.

Using the unique encryption key generated in block 430, the block of
plaintext is encrypted in block 440 using a standard block cipher encryption method
25 such as data encryption standard (DES), triple DES, advanced encryption standard

(AES) or other standard block cipher encryption method. The content identification is appended to the resulting ciphertext and the ciphertext and appended content identification are transmitted in block 450 over the unsecured interface for storage on the storage device.

5 **Retrieval and Decryption of Plaintext—Figures 4 and 5:**

Referring to figure 5, when use of the previously encrypted plaintext is required, the ciphertext and appended content identification are retrieved in block 510 from the storage device. Using the appended content identification in conjunction with host identification 202, the encryption key is recreated. Whichever
 10 method was followed to generate the encryption key from a combination of the host identification and the content identification in block 430 for encrypting the plaintext, the same method is used to generate the encryption key in block 530 for decrypting the ciphertext.

As previously discussed, using the same method to combine the host
 15 identification and the content identification to generate the encryption key results in an encryption key that is deterministic. In other words, using the same host identification and the same content identification to generate the encryption key will always produce in the same encryption key. Referring to figures 4 and 5, the encryption keys generated in blocks 430 and 530 are the same encryption keys.
 20 The encryption key generated in block 530 is used in block 540 to decrypt the ciphertext retrieved in block 510.

In the alternative embodiment, the time variable 206 is used to generate the encryption key in blocks 430 and 530 is a time element, such as the month and year. In this embodiment the time variable is not stored with the ciphertext.
 25 Instead, when the ciphertext is decrypted, the same time element is used, the

month and the year in this example. If the month has changed, the encryption key generated in block 530 will not match the encryption key generated in block 430. Thus, the ciphertext cannot be decrypted. Adding the time variable to the present method for encryption key generation prevents a user from retrieving and
5 decrypting outdated information.

An example of a use for an encryption key that expires is video transmission such as pay-for-view. In this example, the ordered digital video content is encrypted using a unique content identification and the host identification that ordered the video. This results in an encrypted video stream that can only be
10 decrypted by the host device, similar to public key encryption. Adding a time variable to the encryption key generation prevents the encrypted video from being decrypted at a later time or from being decrypted by a device other than the specific host device. While the time variable has been described using digital video, the use is for illustration only and not as a limitation. The time variable can also be
15 used for securing audio content, digital files and databases, just to name a few alternative uses.

As to alternative embodiments, those skilled in the art will appreciate that the present method for encryption key generation may be implemented with alternative size variables. While the generation of the encryption has been discussed using
20 one-byte host identification and a one-byte content identification, the size is for illustration. Those skilled in the art of encryption key generation will appreciate that alternative size variables can be substituted. Likewise, although the content identification can be incremented for each successive block of plaintext, alternative methods of modifying or creating a new content identification for each successive
25 block of plaintext can be substituted.

It is apparent that there has been described a method for encryption key generation that fully satisfies the objects, aims, and advantages set forth above. While the method for encryption key generation has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, 5 modifications, and/or variations can be devised by those skilled in the art in light of the foregoing description. Accordingly, this description is intended to embrace all such alternatives, modifications and variations as fall within the spirit and scope of the appended claims.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2